

200400479-2

10/827,523

IN THE CLAIMS:

The status and content of each claim follows.

1. (currently amended) A method of translating binary code instructions from a source format to a target format for processing by a target processor, said method comprising the steps of:

identifying a source instruction;

selecting a translation template corresponding to said identified source instruction, said template providing a set of target instructions semantically equivalent to said identified source instruction;

translating said identified source instruction in accordance with said template, wherein said translating comprises:

converting said source instruction into a source intermediate data structure having a plurality of members;

mapping said members in said source intermediate data structure to corresponding members in a target intermediate data structure according to said template; and

converting said target intermediate data structure into a target instruction; and
generating dependency analysis code using input and output resources named in the template, the generating of dependency analysis code including:

maintaining a counter associated with each instruction to indicate a group number to which the instruction belongs, wherein all instructions of a same group are issued in parallel; and

assigning each instruction to an earliest group of instructions such that all producers of input and output resources of the instruction have already been assigned to previous groups; and

outputting said ~~translated~~ target instruction for processing by said target processor.

2. (original) A method according to claim 1 in which said source and target instructions include a control part and a data part and said control part being used in said identification step to identify an instruction.

200400479-2

10/827,523

3. (currently amended) A method according to claim 2 ~~in further comprising a transformation step in which~~ wherein said data part from said source instruction is transformed into said corresponding data part or parts of said set of target instructions.

4. (original) A method according to claim 3 in which said transformation step is carried out in accordance with a bit filling routine associated with said template.

5. (original) A method according to claim 4 in which said bit filling routine is uniquely associated with said template.

6. (original) A method according to claim 3 in which said transformation step is arranged to transform data of one type of endianness to data of another type of endianness.

7. (original) A method according to claim 2 in which said source instruction control parts are each concatenated to provide a unique identifier and said templates are indexed in accordance with said identifiers.

8. (original) A method according to claim 7 in which said templates are indexed by said unique identifiers in a look up table.

9. (original) A method according to claim 1 in which said translation is carried out at runtime of an emulated application program.

10. (original) A method according to claim 1 in which said templates are provided by software procedure calls.

11. (original) A method according to claim 1 in which said source format is 32 bit and said target format is 64 bit.

12. (original) A method according to claim 1 in which said source format is PA-RISC code and said target format is ItaniumTM code.

200400479-2

10/827,523

13. (currently amended) An apparatus for translating binary code instructions from a source format to a target format for processing by a target processor, the apparatus comprising:

at least one processor configured to execute code embodied on a computer readable medium;

an instruction identifier embodied within said code for identifying a source instruction;

a template selector embodied within said code for selecting a translation template corresponding to said identified source instruction, said translation template comprising a set of target instructions semantically equivalent to said identified source instruction and further comprising input and output resources; and

a translator embodied within said code for translating said identified source instruction in accordance with said template, wherein said translator is configured to convert said source instruction into a source intermediate data structure having a plurality of members;

map said members in said source intermediate data structure to corresponding members in a target intermediate data structure according to said template; and

convert said target intermediate data structure into a target instruction; and

a scheduler that performs dependency analysis using a counter associated with each instruction to indicate a group number to which the instruction belongs, wherein the counter computes the earliest group to which a given instruction can be assigned such that all producers of inputs and outputs of the instruction have already been assigned to previous groups;

an analysis routine generator that generates dependency analysis code using the dependency analysis and

an output buffer embodied within said code for outputting said translated target instruction for processing by said target processor.

14. (original) Apparatus according to claim 13 in which said source and target instructions include a control part and a data part and said instruction identifier uses said control part to identify an instruction.

200400479-2

10/827,523

15. (currently amended) Apparatus according to claim 14 in which in said translator is operable to transform said data part from said source instruction into said corresponding data part or parts of said set of target instructions.

16. (original) Apparatus according to claim 15 in which said transformation is carried out in accordance with a bit filling routine associated with said template.

17. (original) Apparatus according to claim 16 in which said bit filling routine is uniquely associated with said template.

18. (currently amended) Apparatus according to claim 15 in which translator is operable to transform data of one type of endianness into data of another type of ~~end ianness~~ endianness.

19. (original) Apparatus according to claim 14 in which said source instruction control parts are concatenated to provide a unique identifier and said templates are indexed in accordance with said identifiers.

20. (original) Apparatus according to claim 19 in which said templates are indexed by said unique identifiers in a look up table.

21. (original) Apparatus according to claim 13 in which said translation is carried out at runtime of an emulated application program.

22. (original) Apparatus according to claim 13 in which said templates are provided by software procedure calls.

23. (original) Apparatus according to claim 13 in which said source code has a 32 bit format and said target code has a 64 bit format.

24. (original) Apparatus according to claim 13 in which said source code is PA-RISC code and said target code is Itanium™ code.

200400479-2

10/827,523

25. (currently amended) A computer program product for translating binary code instructions from a source format to a target format for processing by a target processor, comprising a computer-readable medium, further comprising:

a template embodied within said computer-readable medium for use in a binary code translator for translating binary code instructions from a source format to a target format for processing by a target processor, the template comprising:

a template identifier for uniquely associating said template to a source instruction;

a set of target instructions in a target format semantically equivalent to the source instruction;

computer usable program code embodied within said computer-readable medium configured to convert said source instruction into a source intermediate data structure having a plurality of members;

computer usable program code embodied within said computer-readable medium configured to map said members in said source intermediate data structure to corresponding members in a target intermediate data structure according to said template; and

computer usable program code embodied within said computer-readable medium configured to convert said target intermediate data structure into a target instruction.

~~a set of codes for performing dependency analysis using a counter associated with each instruction to indicate a group number to which the instruction belongs, wherein the counter computes the earliest group to which a given instruction can be assigned such that all producers of inputs and outputs of the instruction have already been assigned to previous groups; and~~

~~a set of codes for generating dependency analysis code using the dependency analysis.~~

26. (previously presented) A computer product according to claim 25, further comprising a set of codes causing a computer to derive the template identifier from a control part of the source instruction.

200400479-2

10/827,523

27. (previously presented) A computer product according to claim 26, wherein the template causes a computer to transform a data part of the source instruction into at least one corresponding data part of the set of target instructions.

28. (previously presented) A computer product according to claim 27, further comprising a set of codes for causing a computer to bit fill the data part of the source instruction.

29. (canceled).

30. (previously presented) A computer product according to claim 26, wherein the template causes a computer to create the template identifier by concatenating the control part of said source instruction.

31. (previously presented) A computer product according to claim 25, wherein the template causing a computer to transform a source instruction having a 32 bit format to a target instruction having a 64 bit format.

32. (previously presented) A computer product according to claim 25, wherein the template causes a computer to transform PA-RISC source code into Itanium™ target code.

33. (currently amended) A computer program product for translating binary code instructions from a source format to a target format for processing by a target processor, comprising:

a computer-readable medium, comprising:

a first set of codes for causing a computer to identify a source instruction;

a second set of codes for causing a computer to select a translation template corresponding to said identified source instruction, said template providing a set of target format instructions semantically equivalent to said identified source instruction;

a third set of codes for causing a computer to translate said identified source instruction in accordance with said template by converting said identified source instruction to an intermediate source data structure, mapping members of said intermediate source data

200400479-2

10/827,523

structure to a members of a target intermediate data structure according to said template, and converting said target intermediate data structure into a target instruction; and

~~a fourth set of codes for performing dependency analysis using a counter associated with each instruction to indicate a group number to which the instruction belongs, wherein the counter computes the earliest group to which a given instruction can be assigned such that all producers of inputs and outputs of the instruction have already been assigned to previous groups;~~

~~a fifth set of codes for generating dependency analysis code using input using the dependency analysis; and~~

a ~~sixth~~ fourth set of codes for causing a computer to output said translated instruction for processing by said target processor.

34-35. (canceled).